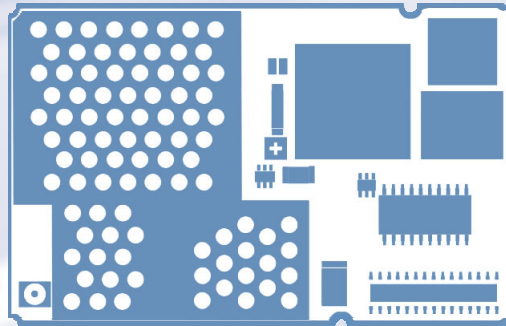


SIEMENS



Upgrading Firmware on MC75, TC6x, AC75

Siemens Cellular Engine

Version: v04

DocID: MC75_TC6x_AC75_AN_16_FW_Update_v04

Supported Products: MC75, TC63, TC65, TC65 Terminal, AC75

Application Note 16

Application Note 16: **Upgrading Firmware on MC75, TC6x, AC75**

Supported Products **MC75, TC63, TC65, TC65 Terminal, AC75**

Version: **v04**

Date: **June 01, 2006**

DocId: **MC75_TC6x_AC75_AN_16_FW_Update_v04**

Status: **Confidential / Released**

General Notes

Product is deemed accepted by Recipient and is provided without interface to Recipient's products. The documentation and/or Product are provided for testing, evaluation, integration and information purposes. The documentation and/or Product are provided on an "as is" basis only and may contain deficiencies or inadequacies. The Documentation and/or Product are provided without warranty of any kind, express or implied. To the maximum extent permitted by applicable law, Siemens further disclaims all warranties, including without limitation any implied warranties of merchantability, completeness, fitness for a particular purpose and non-infringement of third-party rights. The entire risk arising out of the use or performance of the Product and documentation remains with Recipient. This Product is not intended for use in life support appliances, devices or systems where a malfunction of the product can reasonably be expected to result in personal injury. Applications incorporating the described product must be designed to be in accordance with the technical specifications provided in these guidelines. Failure to comply with any of the required procedures can result in malfunctions or serious discrepancies in results. Furthermore, all safety instructions regarding the use of mobile technical systems, including GSM products, which also apply to cellular phones must be followed. Siemens or its suppliers shall, regardless of any legal theory upon which the claim is based, not be liable for any consequential, incidental, direct, indirect, punitive or other damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information or data, or other pecuniary loss) arising out the use of or inability to use the Documentation and/or Product, even if Siemens has been advised of the possibility of such damages. The foregoing limitations of liability shall not apply in case of mandatory liability, e.g. under the German Product Liability Act, in case of intent, gross negligence, injury of life, body or health, or breach of a condition which goes to the root of the contract. However, Claims for Damages arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on liability for injury of life, body or health. The above provision does not imply a change on the burden of proof to the detriment of the Recipient. Subject to change without notice at any time. The interpretation of this general note shall be governed and construed according to German law without reference to any other substantive law.

Copyright

Copying of this document and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights reserved in the event of grant of a patent or the registration of a utility model or design.

Copyright © Siemens AG 2006

Contents

0	Document History	4
1	Overview	5
1.1	Supported Products	5
1.2	Related Documents	5
1.3	Abbreviations	6
1.4	General Requirements and Precautions	6
1.5	User Defined Settings	6
2	Generic Update Program gWinSwup	7
2.1	Connecting the ME	7
2.2	Download via Serial Interface with IGT and DTR Connected (ASC0)	8
2.3	Download via Serial interface with IGT and DTR Not Connected (ASC1)	9
2.3.1	Baudrate Settings on ASC0 / ASC1	10
2.4	gWinSwup Download via USB	11
2.5	gWinSwup Settings for Repairing Invalid Firmware	12
2.6	Troubleshooting	13
3	Generic Firmware Update	17
3.1	Format of the Firmware File Supplied from Siemens	17
3.2	Requirements for the Firmware Download Utility	17
3.2.1	Known Limitations	18
3.2.2	Download Protocol for ASC0 / ASC1	19
3.2.2.1	Example of Firmware Download Program for ASC0/ASC1	20
3.2.3	Download Protocol for USB	21
3.2.3.1	Example of Firmware Download Program for USB	23
3.3	Error Handling for Generic Firmware Update	25
4	Auxiliary Equipment for Software Download	26
4.1	Download Cable	26

Figures

Figure 1:	gWinSwup download via ASC0	8
Figure 2:	gWinSwup waiting for restart	9
Figure 3:	gWinSwup closes without restoring baudrate	9
Figure 4:	Different baudrate settings of module and gWinSwup	10
Figure 5:	gWinSwup download via USB	11
Figure 6:	Format of *.USF firmware file	17
Figure 7:	Download protocol for ASC0 / ASC1	19
Figure 8:	Download protocol for USB	22
Figure 9:	Schematics of download cable	26
Figure 10:	Practical sample of download cable	26

Tables

Table 1:	Abbreviations	6
Table 2:	Required lines	7
Table 3:	Answer patterns required for generic download utility	18

0 Document History

New document: AN 16, "Upgrading Firmware on MC75, TC6x, AC75", Version **v04**

Preceding document: AN 16, " Upgrading Firmware on MC75, TC6x, AC75", Version 03

Chapter	What is new
4.1	Changed capacitor for IGT-DTR connection (now 4.7µF) in Figure 10.

New document: AN 16, "Upgrading Firmware on MC75, TC6x, AC75", Version v03

Preceding document: AN 16, " Upgrading Firmware on MC75, TC6x, AC75", Version 02

Chapter	What is new
1.4	Shortened description as requirements are detailed in chapters 2 - 2.6 and 3.3.
2 - 2.6	Enhanced all chapters, added new chapters to describe differences depending on connection between IGT and DTR. Added further details on baudrate settings.

New document: AN 16, "Upgrading Firmware on MC75, TC6x, AC75", Version v02

Preceding document: AN 16, " Upgrading Firmware on MC75, TC6x, AC75", Version 01

Chapter	What is new
Throughout document	Complete revision because document relates only to gWinSwup and Generic Update. Deleted description of other download programs. Added AC75.

1 Overview

The firmware of the Siemens wireless modules is stored in a Flash memory. To meet the requirements of different application design strategies firmware updates can be downloaded via the interfaces ASC0, ACS1 and USB.

For each product, firmware updates are made available as a binary file *.usf and as an executable program referred to as gWinSwup. The binary file is intended for download through the host application or a dedicated program developed by the customer for any operating system, referred to as generic update. The executable program integrates the *.usf binary file and is ready for use on systems running Windows XP or Windows 2000.

Note: Products with SPI interface, such as AC75 Module, TC65 Module and TC65 Terminal, also support upgrading firmware over the air. For details please refer to the "Application Note 17: Firmware Update Over the Air".

1.1 Supported Products

This Application Note applies to the following Siemens products:

- MC75 Module
- TC63 Module
- TC65 Module
- TC65 Terminal
- AC75 Module

Throughout this document, the abbreviation "ME" is used to refer to all supported products.

The supplied firmware update files are specific to each product type. File names include the product name and the release number (e.g. *gswup_MC75_<0300>.exe* and *MC75_<0300>.usf*). Throughout this document these are replaced with variables (e.g. *gswup_<xyyy>_<release>.exe* and *<xyyy>_<release>.usf*).

1.2 Related Documents

- [1] Hardware Interface Description related to your Siemens product
- [2] AT Command Set related to your Siemens product
- [3] DSB75 Support Box - Evaluation Kit for Siemens Cellular Engines
- [4] Application Note 17: Over-The-Air Firmware Update

To visit the Siemens Website you can use the following link:

<http://www.siemens.com/wm>

1.3 Abbreviations

Table 1: Abbreviations

Abbreviation	Description
ASC0 / ASC1	Asynchronous Controller. Abbreviations used for the first and second serial interface of the supported products. TC65 Terminal has only the ASC0 interface (labeled "Serial" on the casing).
B2B	Short for board-to-board connector
COM	Serial port
DSB	Development Support Box
DTR	Data Terminal Ready
IGT	Ignition
ME	Mobile Equipment Throughout this document, short for all supported products listed in Chapter 1.1.
RXD	Receive direction
SWUP	Software update process
TXD	Transmit direction
URC	Unsolicited Result Code
USB	Universal Serial Bus Not available on TC65 Terminal.

1.4 General Requirements and Precautions

- The firmware download is only intended to install the same or a new firmware version. Trying to downgrade to an older version will be denied.
- Take every precaution to avoid disruption of the firmware download. Do not bend, stress or remove any cable. In the event of failure, there would be no valid software installed - although the update process can be started again any time.
- See Chapters 2.6 and 3.3 for troubleshooting specific problems.

1.5 User Defined Settings

The download procedure has no impact on user defined parameters. All non-volatile user settings will be preserved.

2 Generic Update Program gWinSwup

gWinSwup (*gwsup_<xyy>_<release>.exe*) is a ready-to-use program for Windows XP and Windows 2000 designed to transfer the firmware from a Windows computer to the module's flash memory.

In the following sections the DSB75 Evaluation Board is used as an example to explain technical requirements and procedures involved in the usage of gWinSwup.

2.1 Connecting the ME

Depending on the selected interface, gWinSwup uses the following lines:

Table 2: Required lines

Used ME interface	Lines used on ME side	Lines used on application side (e.g. PC)
ASC0 / ASC1	RXD0, TXD0, IGT, GND	RXD, TXD, DTR connected to IGT of ME, GND
USB	IGT, USB_DP, USB_DN, VUSB_IN	Data, power supply, GND lines of USB

Connecting the module

- The ME interface used for the download (ASC0, ASC1 or USB) shall connect to an appropriate port (COMx, USB) of the computer where the gWinSwup software is located.

Furthermore, when downloading firmware over ASC0 or ASC1 it is recommended to have a connection between the ME's IGT line and the computer's DTR line. This enables gWinSwup to start the ME in the following cases: a) at the beginning of the download if the ME is still in Power Down mode, b) after successful firmware transfer, to restore the original baudrate of the ME if the selected gWinSwup baudrate was not identical with the ME's original baudrate (AT+IPR).

If a) and b) are not necessary, because the ME is switched on before gWinSwup starts and because the baud rates selected with gWinSwup and AT+IPR are the same, there is no need to toggle IGT during the download process, and thus no need to connect both lines.

On the DSB75 Evaluation Board, the connection between IGT and DTR is implemented for the first serial interface. So, the DSB75 is well suited for downloading the firmware over the ME's ASC0 interface. All you need to do is to plug the serial cable and the power supply (9-15 Volts) for powering the DSB75 and the connected ME.

Another solution is building a specialized download cable as suggested in Chapter 4.1. Design recommendations for connecting DTR and IGT can be found in the schematic example of a download cable shown in Figure 10.

For the USB interface, the connection between DTR and IGT is not relevant, but the ME must always be switched on before starting the download.

Connecting the TC65 Terminal

- For the TC65 Terminal no extra hardware is required as its serial port (ASC0 only) can be easily plugged to a free COM port of the computer which holds gWinSwup software.

2.2 Download via Serial Interface with IGT and DTR Connected (ASC0)

- If IGT and DTR are connected the download can be started no matter whether the ME is switched on or off. If the ME is switched on ensure that there is no network activity in progress.
- In any case, close the host application or Terminal program or deactivate it at least (Standby mode) to free the COM port for the download.
- To start the program run the *gwsupw_<xxyy>_<release>.exe* file and select the COM port the module is connected to.
- Select the baudrate. Higher baudrates should be set only if supported by the computer's COM port. See also Chapter 2.3.1 for further detail.
- After the download has completed close gWinSwup and switch on the ME.

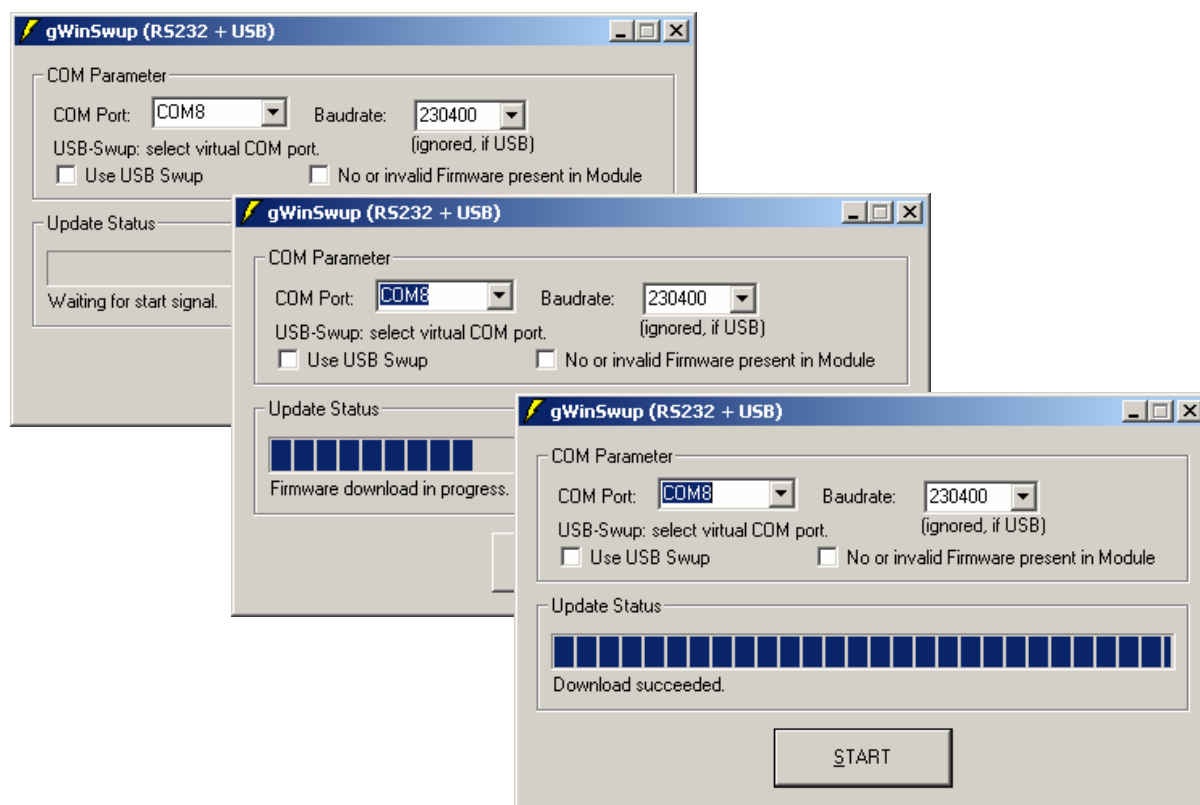


Figure 1: gWinSwup download via ASC0

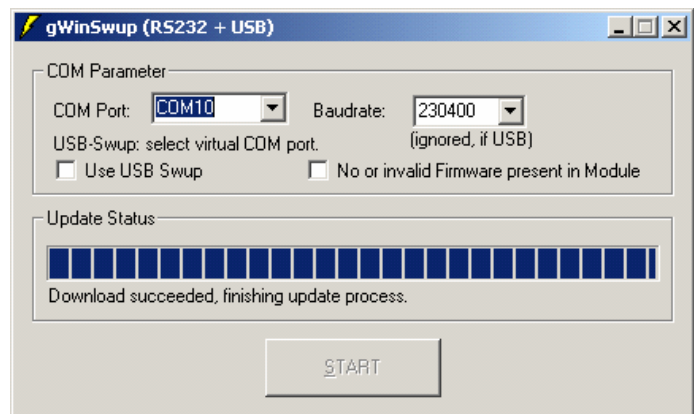
2.3 Download via Serial interface with IGT and DTR Not Connected (ASC1)

The steps are the same as for ASC0, except that gWinSwup does not automatically start the ME if it is powered down. Therefore, before the download starts be sure to switch on the ME and close or deactivate the host application or Terminal program.

The following messages apply only if the baud rate set with gWinSwup differs from the ME's original baud rate set with AT+IPR.

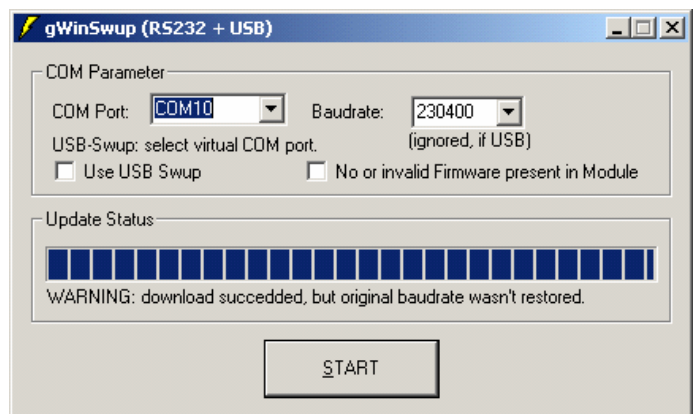
The message given in Figure 2 shows that the transfer was successful, but finally, gWinSwup cannot restart the ME due to lack of the described connection between IGT and DTR. In this case, you have approx. 4 minutes to restart the ME by using the IGT line. This enables gWinSwup to restore the original baudrate.

Figure 2: gWinSwup waiting for restart



If you do not restart the ME via IGT within 4 minutes time gWinSwup confirms that the original baudrate was not restored.

Figure 3: gWinSwup closes without restoring baudrate



2.3.1 Baudrate Settings on ASC0 / ASC1

gWinSwup supports the following baudrates: 57600, 115200, 230400, 460800 Baud. The setting is relevant only for downloading the firmware over ASC0 and ASC1.

The baudrate for the firmware download can be set regardless of the module's baudrate configured with AT+IPR. Thus, you can take advantage of a higher baudrate (if supported by the computer's COM port).

If the baudrate selected for the download differs from the AT+IPR setting, gWinSwup briefly shows the module's original setting before starting the download at the selected baudrate. See example in Figure 4. After successful transfer the original AT+IPR setting of the module is valid again.

If the download aborts prematurely and, after the next attempt, completes successfully, the original baudrate cannot be restored. See issue "Download abort on ASC0 / ASC1" in Chapter 2.6.

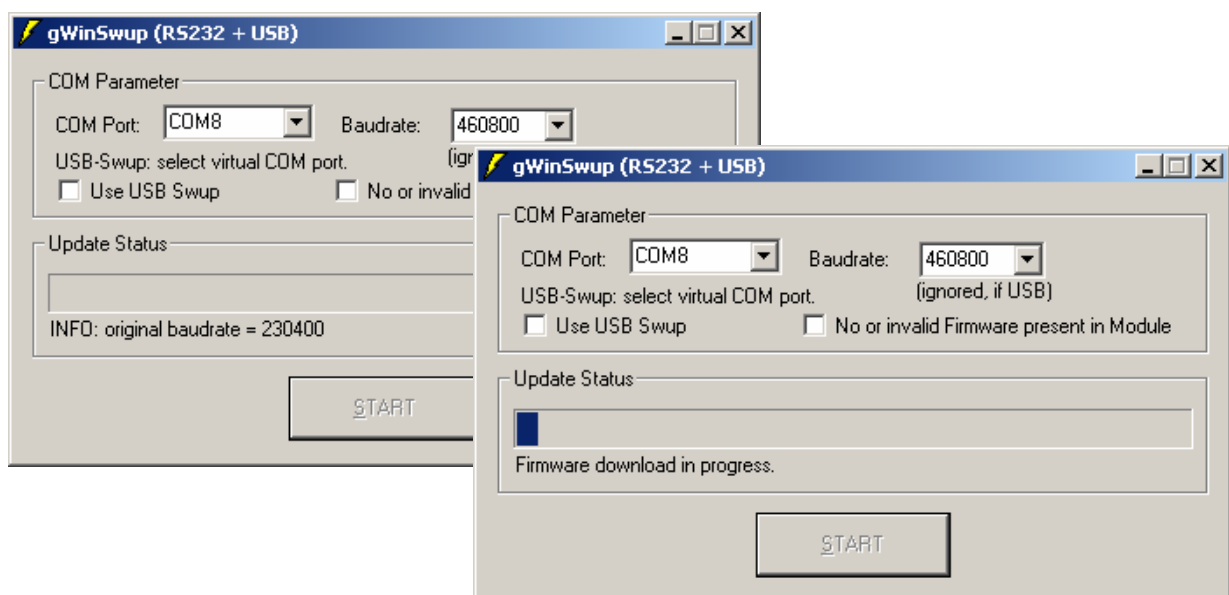


Figure 4: Different baudrate settings of module and gWinSwup

57600 Baud will slow down the download speed and shall be used only if the module has no valid firmware installed. See Chapters 2.5 and 2.6.

2.4 gWinSwup Download via USB

- Do not power down the ME, but close the host application or Terminal program to release the virtual COM port for the download. Ensure that there is no network activity in progress.
- To start the program run the *gwsupw_<xyy>_<release>.exe file*. Choose the virtual COM port assigned to the module's USB interface. Check the option "Use USWB Swup".
- After the download has completed close gWinSwup.
- Switch on the ME by using the IGT line. Then open your host application or Terminal program connected the virtual COM port.

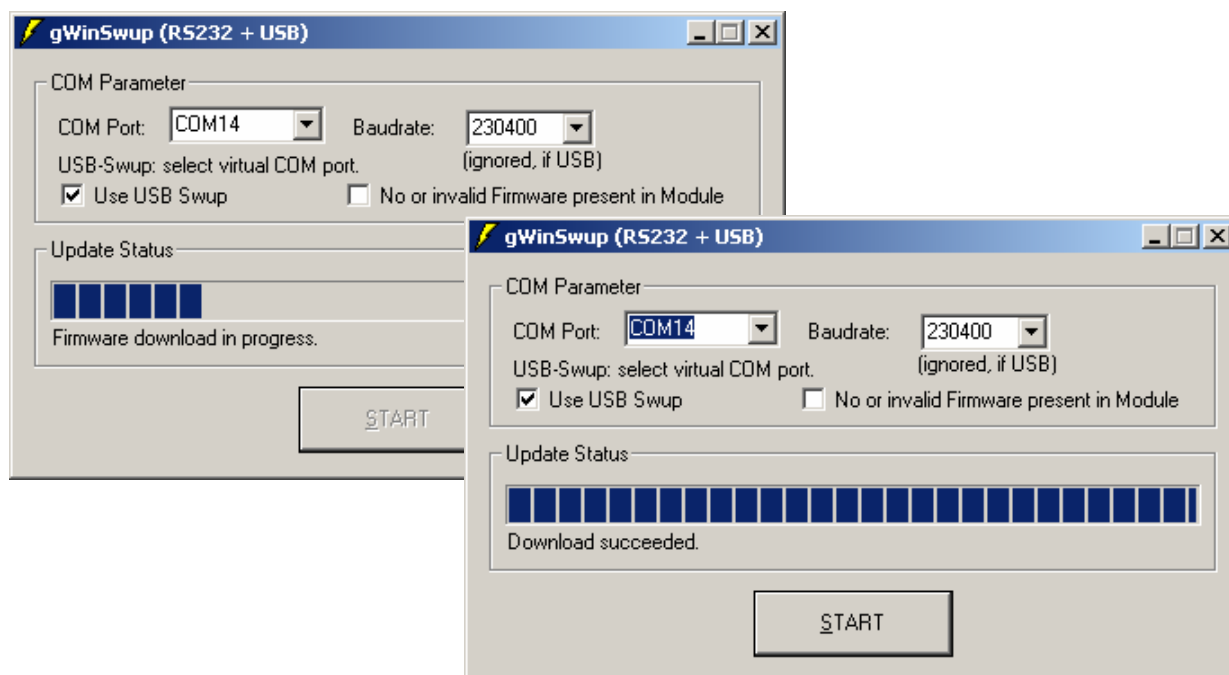


Figure 5: gWinSwup download via USB

2.5 gWinSwup Settings for Repairing Invalid Firmware

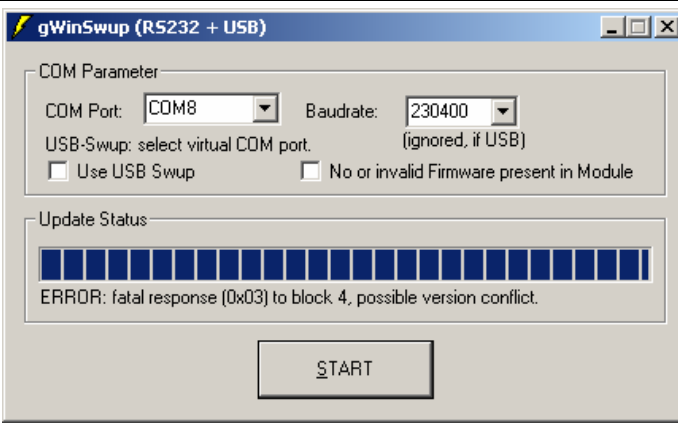
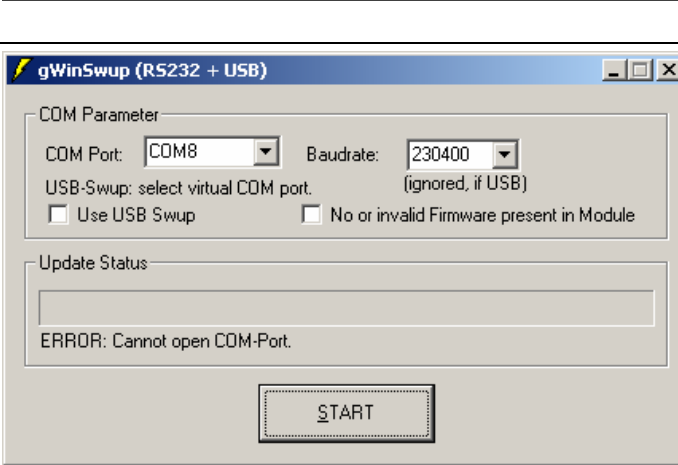
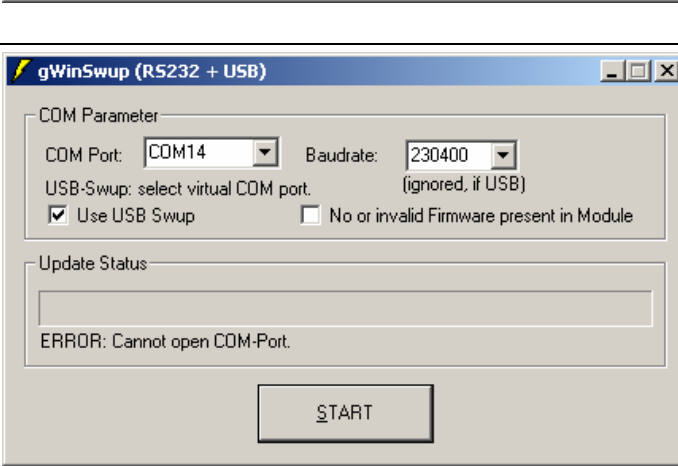
If the ME is switched on although it has no valid software installed or is not responding, it waits 60 seconds for the download to be started. If the download does not begin, the ME switches off after the timeout.

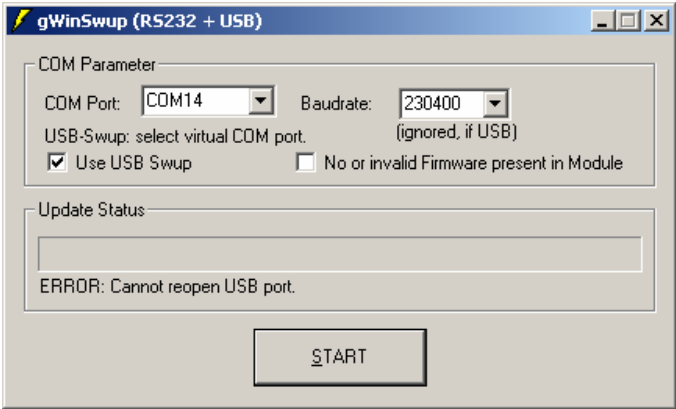
After starting gWinSwup check the option “No or invalid Firmware present in Module”. If the option is activated on ASC0 or ASC1 the gWinSwup program automatically reduces its baudrate to 57600 Baud. As a result, the download takes more time. If the option is checked on USB the download takes the same time as under normal conditions.

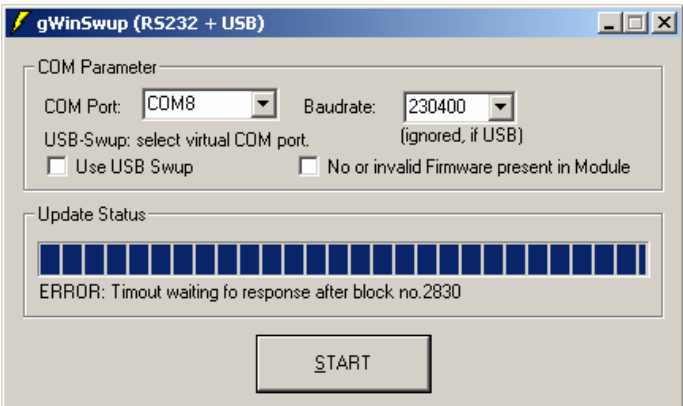
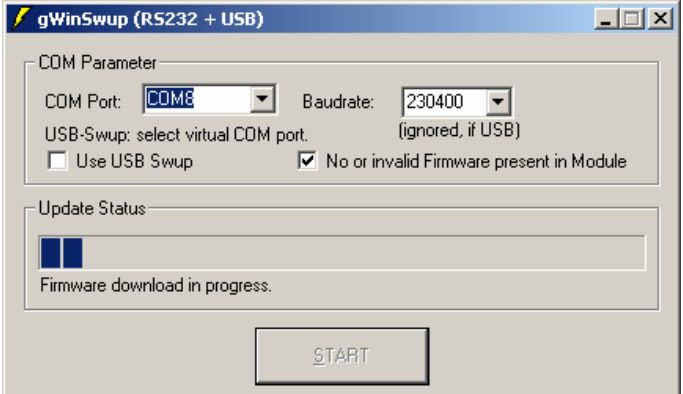
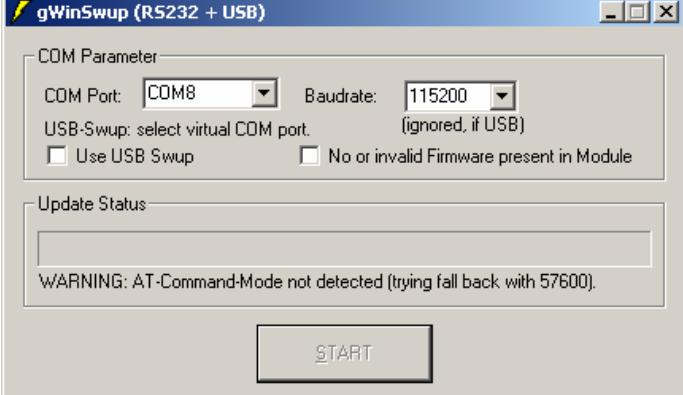
Do not check the option “No or invalid Firmware present in Module” if the ME is working normally. If you do so, gWinSwup will return an error message. The ME needs to be rebooted.

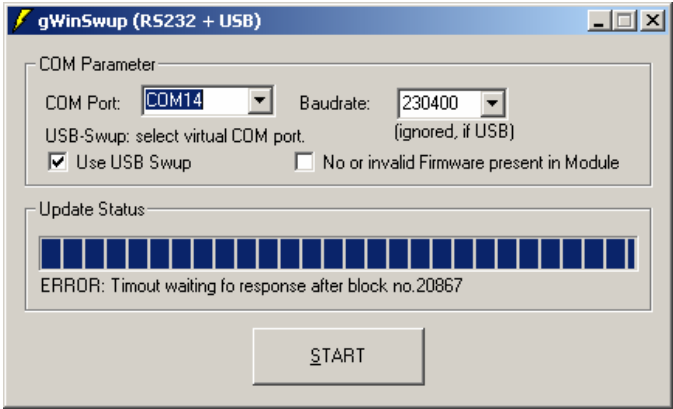
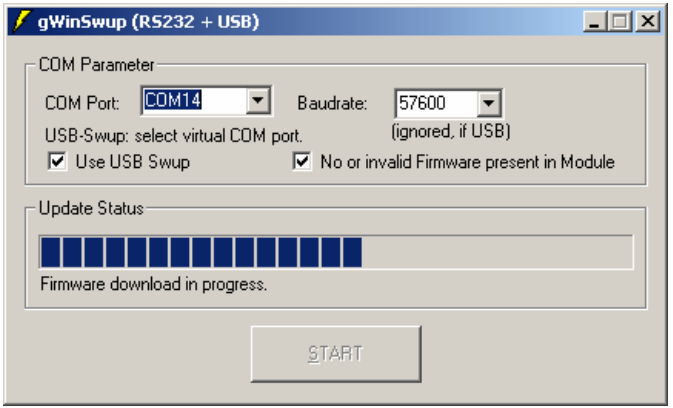
See also Chapters 2.6 and 3.3.

2.6 Troubleshooting

gWinSwup dialog box	Meaning
	<p>Wrong firmware:</p> <p>You tried to install an older firmware version or the firmware of another Siemens module type. The ME instantly switches off.</p>
	<p>Download failure on ASC0 / ASC1:</p> <ol style="list-style-type: none"> The COM port is not free for the download. Probably, the host application or the Terminal program is still open. The baudrate selected with gWinSwup is higher than the maximum rate supported by the computer's COM port.
	<p>Download failure on USB:</p> <p>Probably the ME was powered down. Restart the ME and close the host application or Terminal program. Then restart the firmware download.</p>

gWinSwup dialog box	Meaning
	<p>All products:</p> <p>a) Download failure on USB because the ME is still active on another interface, e.g. ASC0 / ASC1. Close all programs connected to ASC0 / ASC1. Ensure that only the USB interface is active and the virtual COM connected to the USB communicates with the Windows environment. The host application or Terminal program at the virtual COM port shall be closed as well.</p> <p>MC75, AC75 only:</p> <p>b) The error may also occur if the USB parameters set with AT^SCFG do not match the USB *.inf file. Ensure that the original usbmodem.inf file supplied by Siemens is available on the host device. Restore the factory defaults of the Siemens USB descriptors set with AT^SCFG, feature "Serial/USB/DDD" feature. See [2] for details on AT^SCFG.</p>

gWinSwup dialog box	Meaning
	<p>Download abort on ASC0 or ASC1:</p> <p>This error message occurs if the download was aborted, e.g. after switching off the ME or disconnecting the power supply. The ME is no longer responding. If started via IGT the ME switches off after approx. 60s.</p> <p>Solution: Restart gWinSwup from ASC0 / ASC1 and check "No or invalid Firmware present in Module" (Solution 1). If the option remains unchecked (Solution 2) gWinSwup will automatically detect that no valid firmware is installed. During the download, the baudrate of the ME falls back to 57600 Baud (the value indicated in the gWinSwup dialog box is ignored). The firmware download takes about 10 minutes and completes successfully as described in Chapter 2.2.</p>
<p>Solution 1:</p> 	<p>Important note: After the successful download, the ME will be set to the baudrate last detected by gWinSwup. However, this is not the original baudrate defined with AT+IPR, but the rate used by gWinSwup at the time the first download was aborted. If, for example, the aborted download procedure was done at 230400 Baud, the ME will, after the successful firmware transfer, also be set to 230400 Baud. To communicate with the ME, the host application must be set to the same baudrate.</p>
<p>Solution 2:</p> 	<p>If the highest rate 460800 Baud was enabled when the download was aborted, but the host application cannot handle 460800 Baud you will need another program to communicate with the ME and to restore the original baudrate. You can use, for example, HyperTerminal that is normally integrated in a standard Windows XP or 2000 system.</p>

gWinSwup dialog box	Meaning
<div data-bbox="188 322 863 728"></div> <div data-bbox="188 792 863 1198"></div>	<p data-bbox="919 322 1420 358">Download abort on USB:</p> <p data-bbox="919 389 1420 627">This error message occurs if the firmware download was aborted, e.g. after switching off the ME or disconnecting the power supply. The ME is no longer responding. If started via IGT the ME switches off after approx. 60s.</p> <p data-bbox="919 658 1420 963">Solution: Restart the ME with IGT. Wait approx. 5s (this time is needed for Windows to enable the virtual COM port). Restart gWinSwup. Check the option “No or invalid Firmware present in Module”. The download process is performed in the same way as described in Chapter 2.2.</p> <p data-bbox="919 994 1420 1198">Note: If the download fails and ends up with “ERROR: Cannot reopen USB port”, you may have started gWinSwup too early. Please try again as described above.</p>

3 Generic Firmware Update

This chapter gives advice on how to develop a utility dedicated for downloading firmware over the serial interfaces ASC0 or ASC1 or the USB interface into the module's flash memory.

Designed according to these guidelines, the gWinSwup program supplied by Siemens is an example of a ready-to-use download program which integrates the firmware *.usf file and meets the requirements specified below.

3.1 Format of the Firmware File Supplied from Siemens

For this download solution, the firmware must be available in the format *.USF. Files for each type of module can be requested from Siemens. The firmware file consists of records to be transferred one by one from the host application to the module.

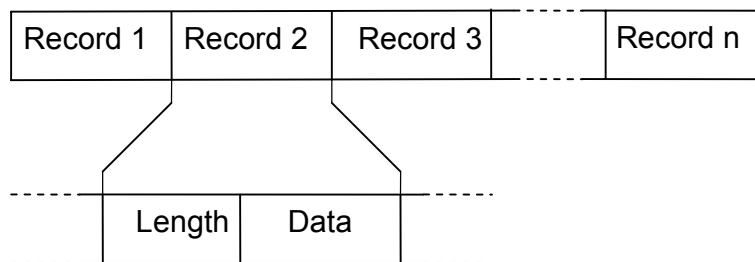


Figure 6: Format of *.USF
firmware file

Length: The 1st byte of each record is the length byte which states the number of data bytes to follow.

Data: 2nd to nth byte

3.2 Requirements for the Firmware Download Utility

The host application or download utility must be designed to meet the requirements of the download process specified in the download protocol. See also Figure 7 and Figure 8 for detail.

First of all, to start the download process, the host application has to send the AT^SFDL command to the module. This causes the module to reset, enter the firmware download mode and send the start pattern to the host application to notify that it is ready to receive records. Upon receipt of the start pattern ("ANSWER_OK" in the example below) the host application must start to send the first record. Optionally, we recommend that the buffer of the serial interface be purged before the module starts to send the first record.

After each record the host application has to wait for an acknowledgment delivered by the module. If this is "OK", the next record can be transferred. "BUSY" means that the module is still processing the record last received – in this case, the transfer must be halted until the next "OK" acknowledgment is received. A checksum verifies the correct transmission of each record. If a download error is detected, the module will send a "RETRY" message.

The host application must be capable to evaluate the answer patterns received from the module and to correctly transmit the data.

Once all records are successfully received, the module switches off automatically. To restart the module use the ignition line IGT.

Appropriate examples are provided in section 3.2.2.1 for ASC0 or ASC1 and in section 3.2.3.1 for USB.

Table 3: Answer patterns required for generic download utility

Answer pattern	Hex value	Function
#define ANSWER_OK	0x01	After the reset caused by AT^SFDL, "ANSWER_OK" notifies the host application that the first record can be sent. After each successfully received record, "ANSWER_OK" notifies the host application that the next data record can be transferred.
#define ANSWER_RETRY	0x02	Notifies that a failure was detected after verifying the checksum. The host application is requested to send the last record again and wait until this record is successfully received and acknowledged by "ANSWER_OK".
#define ANSWER_FATAL	0x03	Download failed at all. Possible causes: The firmware you are trying to download is not intended for this module type, or you are trying to install an older version. The currently installed software is not affected.
#define ANSWER_BUSY	0x04	The module is still processing the record received. The transfer from the host application must be halted until the module sends the next "ANSWER_OK". For example, "ANSWER_BUSY" may appear when the flash is erased on the module. This procedure usually takes a couple of seconds. In this case, "ANSWER_BUSY" is no error.

3.2.1 Known Limitations

- The blocks sent to the module should not really differ in the size of the blocks. Do not use single byte and multiple byte transfer in combination.

3.2.2 Download Protocol for ASC0 / ASC1

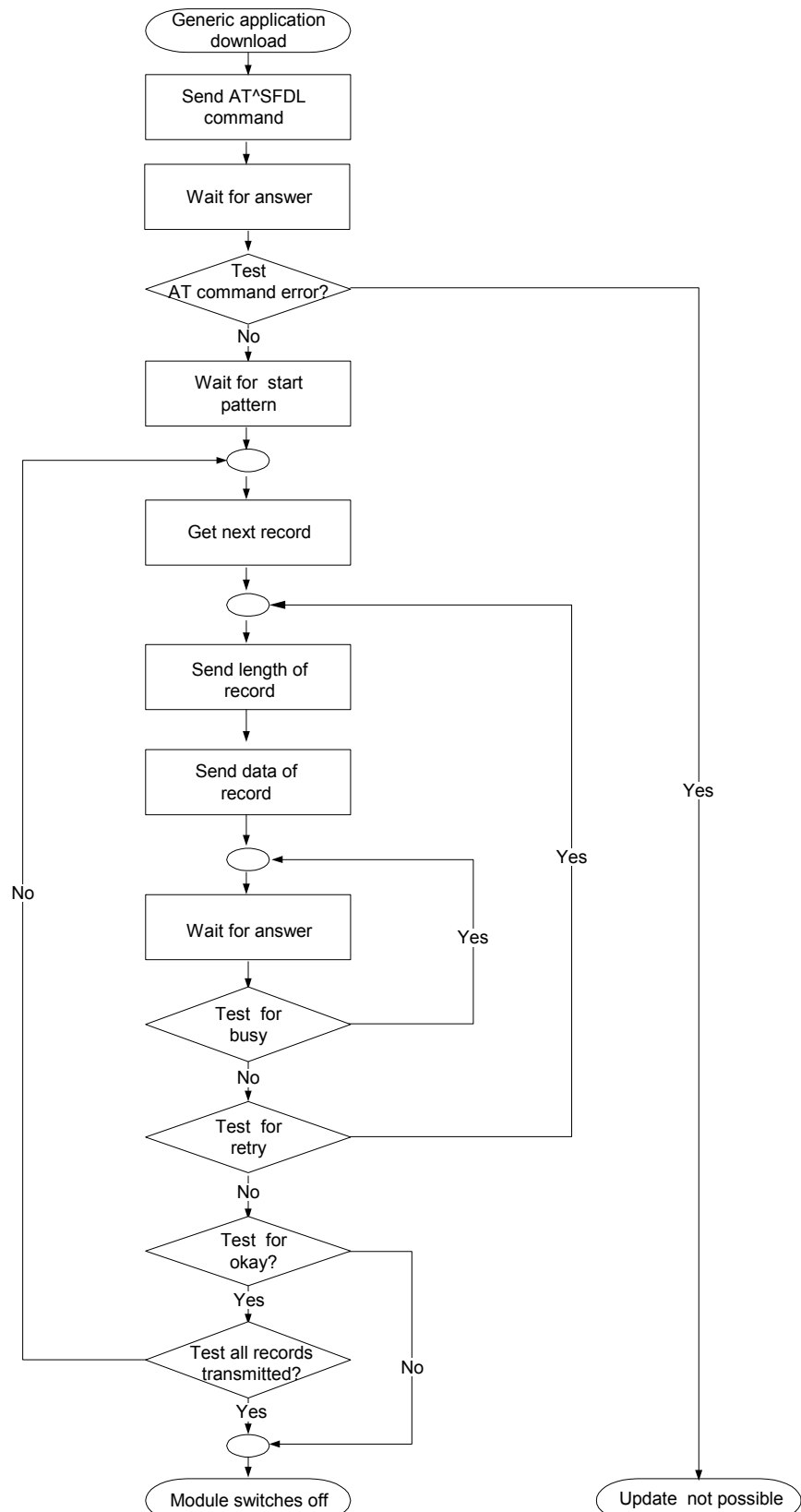


Figure 7: Download protocol for ASC0 / ASC1

3.2.2.1 Example of Firmware Download Program for ASC0/ASC1

This section presents an example of a firmware download utility based on the widely used programming language "C".

```
void DoUpdate (void)
{
    BYTE bAnswer ;
    BYTE bRecordLen ;
    BYTE bRecordData[MAX_RECORD_SIZE] ;

    /* Send AT command */
    WriteToModule ((BYTE *) "AT^SFDL\r", 8) ;

    /* Wait for answer */
    if (!ReadResultLineFromModule ())
        /* Error */
        return ;

    /* Wait for start pattern */
    do
    {
        ReadFromModule (&bAnswer, 1) ;
    }
    while (ANSWER_OK != bAnswer) ;

    do
    {
        /* Read record from file */
        bRecordLen = MAX_RECORD_SIZE ;
        FileReadRecord (&bRecordLen, bRecordData) ;

        do
        {
            /* Send length of record */
            WriteToModule (&bRecordLen, 1) ;

            /* Send data of record */
            WriteToModule (bRecordData, bRecordLen) ;

            do
            {
                /* Wait for answer */
                ReadFromModule (&bAnswer, 1) ;
            }
            while (ANSWER_BUSY == bAnswer) ;
        }
        while (ANSWER_RETRY == bAnswer) ;
    }
    while (ANSWER_OK == bAnswer && !FileEnd ()) ;
}
```

3.2.3 Download Protocol for USB

The firmware download via USB can take place when the module is connected via the USB interface to the USB host. There is no need for a firmware inside a module because the Siemens bootloader includes a fully functional USB stack with a CDC (Communication Device Class) ACM (Abstract Control Model) device implementation.

The module simulates a modem connected via a standard serial COM port. Through this virtual COM port the module can be forced to enter the firmware download mode.

At first the module must be started by an appropriate mechanism, such as asserting the IGT (Ignition) signal or plugging in the USB cable.

Code fragments shown below are in pseudo C code based on the operating system with enabled USB functionality and should be easily adaptable to other operating systems.

What is the major difference between a generic update through ASC0/ASC1 and a generic update through USB?

- The USB scenario requires that the virtual COM port be closed immediately after the AT^SFDL command was sent and OK received. Then, after 2 seconds the COM port shall be opened again.
- The length byte and the data bytes of each record must be transferred in one USB transaction, i.e. by calling the API only once per record.

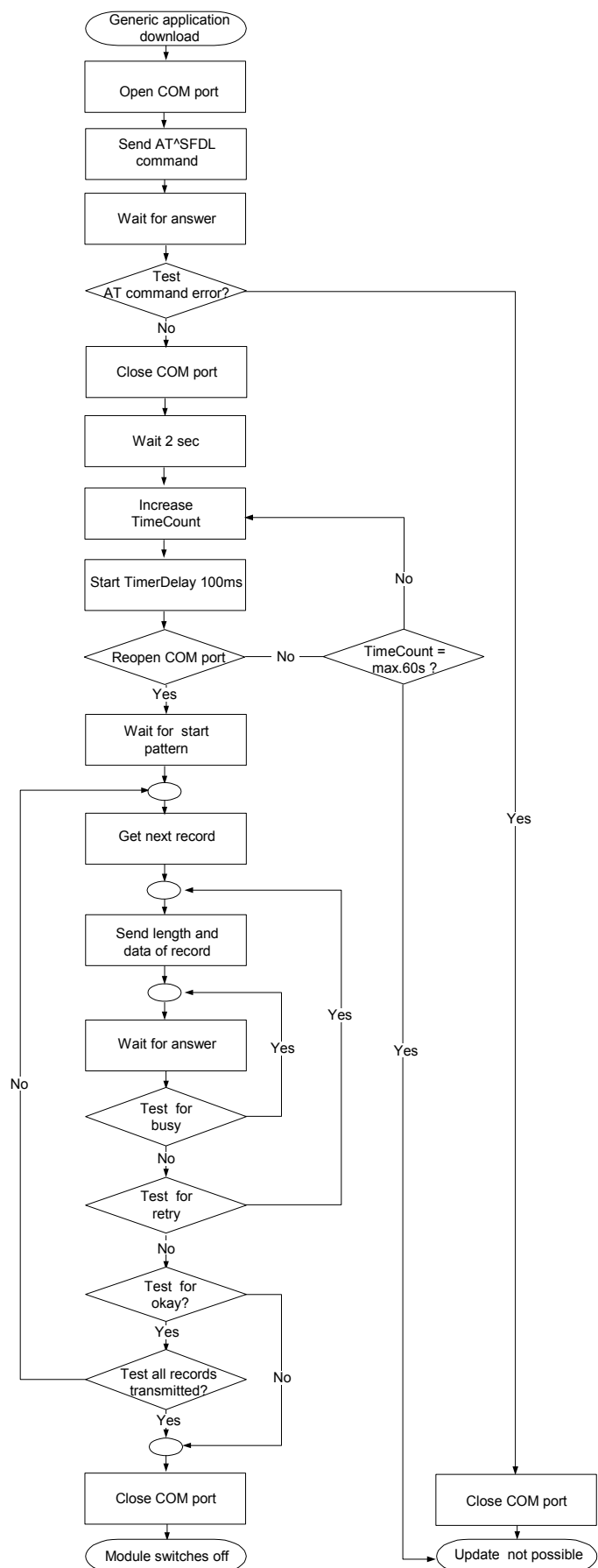


Figure 8: Download protocol for USB

3.2.3.1 Example of Firmware Download Program for USB

```
void DoUpdate (void)
{
    BYTE bAnswer ;
    BYTE bRecord[MAX_RECORD_SIZE] ;
    Int iTry ;          /* Number of tries of open virtual COM port*/

    /* Send AT command */
    WriteToModule ((BYTE *) "AT^SFDL\r", 8) ;

    /* Wait for answer */
    if (!ReadResultLineFromModule ())
        /* Error */
        return ;

    /* Close port because the virtual COM port of the module shall
       be disabled for a short time */
    CloseComPort() ;

    /* Allow USB host to disable the virtual COM port. */
    Wait (2 SECONDS) ;

    /* Try to reopen the virtual COM port again */
    iTry = 0 ;
    do
    {
        if (OpenComPort ())
            /* Virtual COM port is opened again */
            iTry = -1 ;
        else
        {
            /* Virtual COM port is not opened, wait 100 ms */
            Wait (100 MILLISECONDS) ;
            iTry++ ;
        }
    }
    while (0 <= iTry && 80 > iTry) ;
    if (-1 != iTry)
        /* Error, virtual COM port is not opened again */
        return ;

    /* Wait for start pattern */
    do
    {
        ReadFromModule (&bAnswer, 1) ;
    }
    while (ANSWER_OK != bAnswer) ;

    do
    {
        /* Read record from file */
        bRecord[0] = MAX_RECORD_SIZE ;
        FileReadRecord (bRecord, bRecord + 1) ;
    }
```

```
do
{
    /* Send length and data of record */
    WriteToModule (bRecord, bRecord[0] + 1) ;

    do
    {
        /* Wait for answer */
        ReadFromModule (&bAnswer, 1) ;
    }
    while (ANSWER_BUSY == bAnswer) ;
}
while (ANSWER_RETRY == bAnswer) ;
}
while (!FileEnd () && ANSWER_OK == bAnswer) ;
}
```


3.3 Error Handling for Generic Firmware Update

1. If a valid firmware is installed, but the AT^SFDL command is issued without sending any firmware afterwards the module will stay in the download mode. In this case it must be switched off.
2. Disconnecting the power supply during a firmware download should be strictly avoided. If nevertheless the download is aborted due to power failure, the module will be left in an undefined state with no valid software installed. On the ASC0 / ASC1 interfaces, the module's baudrate will be set to 57600 Baud in this case.
After reconnecting power and restarting the module, you have only 1 minute to enter the AT^SFDL command and retry the download from one of the interfaces ASC0, ASC1 or USB. Also note that no other AT commands will be accepted. If the download fails to start within this time the module switches off again.

See also Chapter 2.6 for troubleshooting common gWinSwup problems.

4 Auxiliary Equipment for Software Download

4.1 Download Cable

This section provides the schematics of a universal RS232 download cable usable for all types of Siemens GSM modules and all download options described in this document for ASC0 and ASC1.

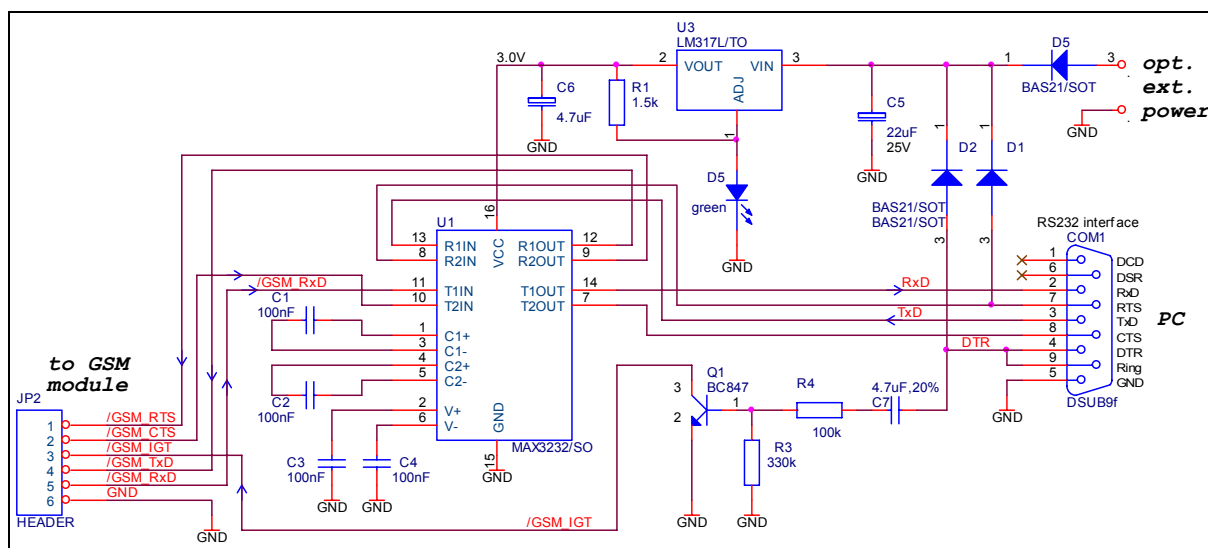


Figure 9: Schematics of download cable

The box shown in Figure 10 is an example made of commercially available components. It is not available as a Siemens product.



Figure 10: Practical sample of download cable